Iterative Search Attribution for Deep Neural Networks

Zhiyu Zhu [1] **Huaming Chen** [1] **Xinyi Wang** [2] **Jiayu Zhang** [3] **Zhibo Jin** [1] **Jason Xue** [4] **Jun Shen** [5]

Abstract

Deep neural networks (DNNs) have achieved state-of-the-art performance across various applications. However, ensuring the reliability and trustworthiness of DNNs requires enhanced interpretability of model inputs and outputs. As an effective means of Explainable Artificial Intelligence (XAI) research, the interpretability of existing attribution algorithms varies depending on the choice of reference point, the quality of adversarial samples, or the applicability of gradient constraints in specific tasks. To thoroughly explore the attribution integration paths, in this paper, inspired by the iterative generation of high-quality samples in the diffusion model, we propose an Iterative Search Attribution (ISA) method. To enhance attribution accuracy, ISA distinguishes the importance of samples during gradient ascent and descent, while clipping the relatively unimportant features in the model. Specifically, we introduce a scale parameter during the iterative process to ensure the features in next iteration are always more significant than those in current iteration. Comprehensive experimental results show that our method has superior interpretability in image recognition tasks compared with state-of-the-art baselines. Our code is available at: https://github.com/LMBTough/ISA

1. Introduction

DNNs have achieved state-of-the-art performance in the tasks of computer vision (Esteva et al., 2021; Jabbar et al., 2018; Pathak et al., 2018), natural language processing (Collobert & Weston, 2008; Ozcan et al., 2021; Rajendran & Topaloglu, 2020), and speech recognition (Pan et al., 2012; Maas et al., 2017) and so on. Nevertheless, DNNs are considered as black-box approach impeding human comprehension, which may lead to decision-making errors. Concerning AI safety issue, it becomes essential to provide the interpretability of these models. However, explaining the intermediate processes of model inputs to outputs poses challenges due to the complexity of nonlinear layers and huge number of parameters in DNNs (Pan et al., 2021).

Gradient-based attribution methods are widely used in Explainable Artificial Intelligence (XAI) as they offer an effective way to explain deep learning models. Integrated Gradient (IG) (Sundararajan et al., 2017) method proposes the axiomatic theorem of attribution for the first time and uses the reference input as an anchor on the attribution integration path to calculate the importance of input features. Adversarial Gradient Integration (AGI) method (Pan et al., 2021) is proposed to search for the steepest gradient ascent path of adversarial samples, thereby avoiding the impact of invalid reference selection on attribution accuracy. More Faithful and Accelerated Boundary-based Attribution (MFABA) method (Zhu et al., 2023) uses second-order taylor gradient expansion and hessian approximation to obtain more faithful and accelerated attribution results. To summarize, the literature indicates that prior researches to gradient-based attribution method often rely on baseline points selected (Sundararajan et al., 2017; Wang et al., 2022), the quality of constructed adversarial samples (Pan et al., 2021), or specific gradient rules (Kapishnikov et al., 2021; Zhu et al., 2023). The interpretability of existing attribution algorithms varies due to limited exploration of attribution integration paths, leading to less promising performance.

In this paper, we leverage a unique search space to find the most faithful features of the deep learning model, addressing the challenge of parallel processing of model features. Inspired by the diffusion model (Rogers, 2004), we believe that the process of iteratively computing feature importance using autoregressive properties would be more applicable for the derivation of attribution results. We firstly use both gradient ascent and gradient descent to investigate the impact of variations in the original features on the outputs. Additionally, we propose a theorem to distinguish feature importance during feature search. The redundant features

[1] School of Electrical and Computer Engineering, University of Sydney, Sydney, NSW, Australia [2] Faculty of Computer Science & Information Technology, University of Malaya [3] Suzhou Yierqi, Suzhou, China [4] Data61, CSIRO, Sydney, NSW, Australia [5] University of Wollongong, Australia. Correspondence to: Zhiyu Zhu <nevertough@outlook.com>, Huaming Chen <huaming.chen@sydney.edu.au>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

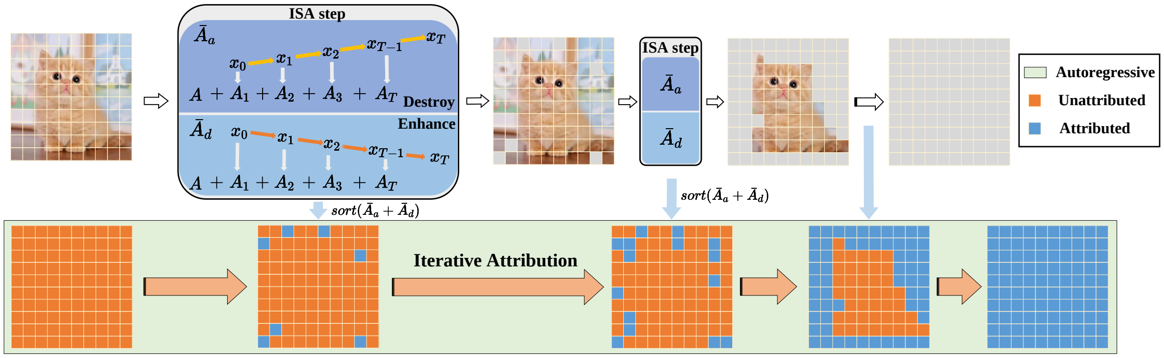
ISA step
$\bar{A}_a$
x_0 x_1 x_2 x_{T-1} x_T
$A + A_1 + A_2 + A_3 + A_T$
Destroy
Enhance
$\bar{A}_d$
x_0 x_1 x_2 x_{T-1} x_T
$A + A_1 + A_2 + A_3 + A_T$
$sort(\bar{A}_a + \bar{A}_d)$
ISA step
$\bar{A}_a$
$\bar{A}_d$
$sort(\bar{A}_a + \bar{A}_d)$
Autoregressive
Unattributed
Attributed
Iterative Attribution

Figure 1. ISA's attribution process (The first row of the image represents the search process of ISA, where each search step includes a Destroy process and an Enhance process to find features that can disrupt/support the model's decision. The second row shows the iterative attribution process, where the blue squares represent regions that have already been attributed, and the orange squares represent regions that are yet to be attributed)

are subsequently clipped by assigning lower attributions. Moreover, we introduce a scale parameter during the iterative process to ensure that the features in next iteration are always more significant than those in current iteration, enhancing the accuracy of feature importance estimation. Figure 1 provides an illustration of our attribution processing step. More details will be provided in Section. 4. The key contributions are summarized as follows:

- We propose a novel iterative attribution method based on gradient ascent and descent search strategies, termed ISA, to improve attribution performance.
- We provide the theoretical proof and in-depth analysis for the ISA method, and perform extensive experiments and ablation study for the evaluation.
- We demonstrate that the ISA method can be easily implemented to obtain SOTA performance in comparison with other attribution methods. The relevant source code is publicly released.

2. Related Work

Current popular methods for interpreting DNNs can be categorised into local approximation and gradient-based attribution methods (Li et al., 2023). For local approximation methods, mostly it can only interpret a single sample or a small portion of samples locally, while for gradient-based attribution methods, it tends to use gradient information to obtain a global interpretation of each feature.

2.1. Local approximation methods

Local approximation methods are dedicated to finding an approximately interpretable surrogate model given diverse model images to compute gradient information and obtain attribution results. The Local Interpretable Model-agnostic Explanations (LIME) algorithm (Ribeiro et al., 2016) combine approximation and weighted sampling methods to construct a local model that gives the interpretable prediction results of the model classifier. Since the LIME algorithm is based on a locally interpretable model, it is less interpretable in the global context and has the potential to produce erroneous results. Shapley Additive Explanations (SHAP) algorithm (Lundberg & Lee, 2017) calculates the contribution of each feature to the prediction result by Shapley value and ranks the importance, thus achieving both local and global interpretation of the model. SHAP algorithm is able to give an adequate explanation of the model prediction in the global context compared to LIME. However, it has a high computational cost due to the need to repeatedly calculate Shapley values for different features. Deep Learning Important Features (DeepLIFT) method (Shrikumar et al., 2017) calculates the importance score of each input feature to explain the prediction effect of the deep learning model. Some other methods including the works in (Datta et al., 2016) and (Fong & Vedaldi, 2017) are also used to obtain the interpretable results for DNNs models.

2.2. Gradient-based attribution methods

In order to obtain reliable evaluations to cope with realistic and highly sensitive deep learning tasks, Grad-CAM (Selvaraju et al., 2017) and Score-CAM (Wang et al., 2020) use gradient information to visualize the contribution values of image pixels to explain the model prediction process. Unfortunately, these two methods are more suitable for CNNs and perform poorly in non-CNN cases (Pan et al., 2021). Saliency Map (SM) method (Simonyan et al., 2013) can obtain interpretable results of non-CNN models, but suffers from gradient saturation and the attribution result may be zero. Layer-wise Relevance Propagation (LRP) (Bach et al., 2015) assigns activation values in the network to input features and calculates each neuron's contribution to the final output through backpropagation. However, LRP is sensitive to small perturbations in the input, affecting the interpretability when facing adversarial samples.

Integrated Gradients (IG) method (Sundararajan et al., 2017) addresses the gradient deficiency of the SM algorithm. By selecting the desired reference points as anchors on a linear integration path, IG integrates the continuous gradients to obtain the attribution of each input feature. However, for the methods like SM, LRP or IG, they are usually considered to be local since they operate based on gradients calculated at specific anchors or input instances.

To explore better anchor selection than IG, Boundary-based Integrated Gradient (BIG) method (Wang et al., 2022) introduces boundary search to obtain more accurate attribution results. Although BIG attempts to use an adversarial sample as anchors, its integration path is still linear. Meanwhile, BIG needs to calculate the gradient of each feature, which increases the computational complexity to some extent. Adversarial Gradient Integration(AGI) method (Pan et al., 2021) is committed to finding a steepest non-linear ascending path from the adversarial example, which does not need reference points on the path like IG. The accuracy of AGI highly depends on the quality of adversarial samples, and the effectiveness changes when the construct method of adversarial samples is varying.

Considering the path noise in IG algorithm, the Guided Integrated Gradients (GIG) method (Kapishnikov et al., 2021) eliminates unnecessary noisy pixel attributions by constraining the network input and back-propagating the gradients of the neurons so that only the pixel attributes associated with the predicted category are retained. However, GIG is limited to the image tasks and the quality of the input features can largely affect the attribution accuracy, while the computational complexity of the algorithm is also an issue. Other variations of IG algorithm such as Fast-IG (Hesse et al., 2021) and Expected Gradient (EG) (Erion et al., 2021) have similar problems.

3. Preliminaries

3.1. Problem definition

Formally, to explain the explicit expression of the DNN model $f(\cdot)$, we define the input feature $x \in R^n$ where n is the dimension of the input feature, and the output of the model $\hat{y} = f(x)$. The goal of attribution is to find $A \in R^n$ to interpret the importance of each feature in x. For easy understanding, we refer to the basic idea of the Saliency Map (Simonyan et al., 2013). If the deep neural network f is continuously differentiable, the input feature importance A of the model will be derived from the gradient information $\frac{\partial f}{\partial x}$. It is important to highlight that this process involves a direct one-to-one mapping. We define the situation where features are unseen to the model as: We change the features to 0, because after the features become 0, the first layer parameters of the model will not be activated.

3.2. Sensitivity and implementation invariance axioms

As mentioned in (Sundararajan et al., 2017), an attribution method satisfies *Sensitivity* if for every input and baseline that differ in one feature but have different predictions then the differing feature should be given a non-zero attribution. For two neural networks with the same inputs and outputs, the attribution is always the same if they satisfy *Implementation Invariance*.

4. Method

4.1. Attribution and Adversarial Attacks

To understand a model's decision-making process, one approach is to identify the minimal feature variations that either disrupt or enhance the current decision. The emphasis on minimality is crucial as changes to a large number of features could sever the semantic link to the original sample. This problem can be reformulated as finding the most impactful feature variations under a constraint of limited changes to disrupt or enhance the model's decision.

In the context of attribution, the gradient $\frac{\partial L(x)}{\partial x}$ w.r.t. the sample x is vital information (Zhu et al., 2023; Pan et al., 2021). Gradients, representing local first-order changes, help evaluate the impact of feature modifications on the model's decisions. However, relying solely on sample's gradient is insufficient (violating the *Sensitivity* axiom). The product of the feature variation value and $\frac{\partial L(x)}{\partial x}$ needs to be introduced (detailed proof in **Appendix. A**).

We introduce the notion of fairness in attribution:

Fairness**:** Each feature should be treated equitably in the attribution process.

Consider a toy example with two one-dimensional features x_1 and x_2, where the change $\Delta x_1 = 1$ and $\frac{\partial L(x_1)}{\partial x_1} = 0.5$, resulting in a product of 0.5. For $\Delta x_2 = 0.1$ and $\frac{\partial L(x_2)}{\partial x_2} = 0.49$, the product is 0.49. Based on the attribution result, x_1 appears more significant. However, x_2, with a change value of 0.1, would be more impactful if its change magnitude was 1, indicating an unfair treatment between x_1 and x_2. Thus, ensuring *Fairness* across different feature dimensions during sample variation and gradient computation is necessary.

Adversarial attacks aim to maximize the loss function with minimal perturbations to the input sample, aligning with the objective of attribution. The use of the sign function in adversarial attacks to decouple feature variations from model parameters (Goodfellow et al., 2014) resonates with our fairness concept, as it results in identical magnitudes of $|\Delta x_i|$. Notably, $\frac{\partial L(x_i)}{\partial x_i} = 0$ rarely occurs in attacks, and even if it does, it signifies that the feature change has a negligible impact on the decision-making process.

Theorem 4.1. *For all* $\Delta x = \{\Delta x_1, \Delta x_2, \ldots, \Delta x_T\}$*, where* $\Delta x_i \in [-1, 0, 1]$*, we have*

$$sign\left(\frac{\partial L(x)}{\partial x}\right) \cdot \frac{\partial L(x)}{\partial x} \geqslant \Delta x \cdot \frac{\partial L(x)}{\partial x} \tag{1}$$

Theorem 4.1 ensures that adversarial attacks seek the optimal feature variations to alter the model's decision-making under *Fairness* concept, with $sign(\frac{\partial L(x)}{\partial x})$ indicating the most effective direction for feature variations.

4.2. Gradient ascent and gradient descent in attribution

We assume that the model input is x, then for the iterative step $t = 0, 1, ..., T$, the model gradient ascent and descent process can be expressed as

$$m = sign(\frac{\partial L(x_t)}{\partial x_t}) \oplus sign(x_t) \tag{2}$$

$$x_t = x_{t-1} \pm \eta \cdot sign(\frac{\partial L(x_{t-1})}{\partial x_{t-1}}) \cdot m \tag{3}$$

The objective of Eq. 2 is to determine the direction of attribution exploration towards removing features. Here, $\oplus$ represents the XOR symbol. $m \in \mathbb{R}^n$ ensures that features will be explored in directions that are unseen to the model by a mask. As defined in Sec. 3.1, exploring features towards 0 signifies exploration in an unseen direction. It is worth noting that, due to the sign function, the processes of gradient ascent and gradient descent will completely explore the entire feature space. $\{\eta, T\} = \{\eta_1, \eta_2, T_1, T_2\}$, which value depends on whether gradient ascent or descent is performed. $\{\eta_1, T_1\}$ is step size and iterative step in gradient ascent, $\{\eta_2, T_2\}$ is step size and iterative step in gradient descent. $\frac{\partial L(x_{t-1})}{\partial x_{t-1}}$ is the derivative of the loss function L w.r.t. the input x_t. It is worth noting that in gradient ascent, ± is a plus sign. Conversely, in gradient descent, ± is a negative sign.

In this work, inspired by AGI (Pan et al., 2021), we design the searching mechanism to identify the integration path instead of the original linear path. We denote x_0 as the original input, the path can be represented as $x_t = x_0 + \sum_{k=0}^{t-1} \triangle x_k$. In order to consider both the role of gradient ascent and gradient descent in feature attribution, we list the attribution steps below.

$$\triangle x_t = \pm\eta \cdot sign(\frac{\partial L(x_t)}{\partial x_t}) \cdot m \tag{4}$$

$$A = \sum_{t=0}^{T-1} \triangle x_t \cdot \frac{\partial L(x_t)}{\partial x_t} \tag{5}$$

Here, $+$ and $-$ represent the operation of gradient ascent and gradient descent, respectively. Our goal is to identify the important features via attribution since attribution can indicate the contribution of each feature to the loss change. Correspondingly, the methods of changing loss include gradient ascent and gradient descent, in which their direction function $sign$ are used in these two processes to update each feature in a fair way. We get attribution A_a during gradient ascent and A_d during gradient descent.

Since the difficulty of feature search corresponding to gradient ascent and gradient descent is different, the value of the loss function changed at the same number of iterations has a variability. We define $\triangle L_a = L(x_{T_1}) - L(x_0)$ and $\triangle L_d = L(x_{T_2}) - L(x_0)$. Here x_{T_1} and x_{T_2} represent the samples in the final stages of gradient ascent and descent, respectively. So we divide the attribution results in Eq. 5 by the total change in the loss function to make the gradient ascent and descent equally competitive. Thus, we get $\bar{A}_a = \frac{A_a}{\triangle L_a} \in [0, 1]$ and $\bar{A}_d = \frac{A_d}{\triangle L_d} \in [0, 1]$ (detailed proof in **Appendix. A**).

Theorem 4.2. *Given a sample* x_0*, where* A_a^i *and* A_a^j *correspond to the attribution values of the* i*-th and* j*-th dimensions after gradient ascent, respectively. If* $A_a^i \geq A_a^j$*, then the feature importance of the* i*-th dimension is greater than that of the* j*-th dimension. This is because if altering a feature increases the loss function (i.e., impairs the model's decision-making), then this feature is important and should not be changed easily. The formula is expressed as:*

$$L(x_0 + \Delta\bar{x_i}) \geq L(x_0 + \Delta\bar{x_j}) \quad s.t. \quad A_a^i \geq A_a^j \tag{6}$$

Here $\triangle\bar{x_i} = [0, ..., \overset{i}{\triangle x_i}, ..., 0]$, $\triangle\bar{x_j} = [0, ..., \overset{j}{\triangle x_j}, ..., 0]$. Similarly in gradient descent, the larger A_d^i represents the weaker ability of the i-th dimension feature to enhance the model (detailed proof in **Appendix. B**).

From *Theorem 4.2*, we can make two *Conclusions*:

1. During gradient ascent, features with *larger* attribution values A_a are important.

2. During gradient descent, features with *smaller* attribution values A_d are important.

Discussion As stated in the *Conclusion*, the *smaller* attribution value in $\bar{A}_a$ is unimportant. Since the sign of ΔL_d is negative, conclusion 2 is transformed into: During gradient descent, the *smaller* attribution value in $\bar{A}_d$ is unimportant.

As $\bar{A}_a$ and $\bar{A}_d$ belong to the same dimensional scale, to make a balance between gradient ascent and descent, we combine them in the following equation (detailed proof in **Appendix. A**):

$$\bar{A} = \bar{A}_a + \bar{A}_d \tag{7}$$

Therefore, the *smaller* attribution value of $\bar{A}$ is *unimportant* in our opinion.

4.3. Prerequisites of ISA

Inspired by diffusion model (Rogers, 2004), attribution can also be converted into an auto-regression process. The final attribution result can be obtained iteratively. We also perform iterative error analysis in the **Appendix. C** to prove the necessity of iterative attribution.

Iterative Integrity Since the attribution values of all features need to be obtained, the iterative process has to iterate over all parameters. Suppose we have n features and each time we attribute k features in x, the total number of iterations is $\Gamma = \lceil \frac{n}{k} \rceil$,$\lceil \cdot \rceil$ denoted as the rounding up function.

Feature removal priority The iterative process requires clipping unimportant features first, because once the most important features are removed, the remaining features will not be enough to support current model decisions, and the importance correlation of these features will become unclear and cannot be further attributed. Therefore, removing unimportant features allows the model to maintain the current decision and continue attribution. Features that are removed first have relatively lower attribution values than those that are removed later. In order to make the attribution results quantifiable, we perform normalization in eq. 9 to ensure that they are between 0 and 1.

$$\bar{a_\gamma} = min_k(\bar{A_\gamma}) \tag{8}$$

$$\bar{a_\gamma} = \frac{\bar{a_\gamma} - min(\bar{a_\gamma})}{max(\bar{a_\gamma}) - min(\bar{a_\gamma})} \tag{9}$$

We get a minimum k number of attribution values in $\bar{A}$ as $\bar{a_\gamma}$, here γ represents the γ-th iteration. Such values are corresponding to the removed features. Following the principle that the early removal features are less important, we need to make sure $max(\bar{a_\gamma}) < min(a_{\bar{\gamma+1}})$. So that we derive

$$\bar{a_\gamma} = \bar{a_\gamma} + \gamma \tag{10}$$

For example, when $\gamma = 0$, $\bar{a_0} \in (0, 1)$. When $\gamma = 1$, $\bar{a_1} \in (1, 2)$. This satisfies the conditions presented above.

4.4. Scaling factor of ISA

We observe that the best results in the previous iteration will perform better than the worst results in the latter iteration. Thus, we add a scaling factor in the iterations to enhance the performance of the algorithm. The specific iterative formulas of ISA are as follows

$$\bar{a_\gamma} = \bar{a_\gamma} \cdot S + \gamma \tag{11}$$

where $S \in [1, 2)$ denotes the scale level of $\bar{a_\gamma}$. So we use an example to explain Eq.11. It is obvious that when $\gamma = 0$, $\bar{a_0} \in (0, S)$. When $\gamma = 1$, $\bar{a_1} \in (1, S+1)$. By analogy, our assumptions are satisfied.

4.5. Axiomatic proof of ISA

Sensitivity During the iteration process, changes in gradient ascent and descent are all captured by original input information. It is also not retroactive because the feature values in previous iterations are invariant in subsequent iterations. Thus, the attribution result must be non-zero.

Implementation Invariance Since our algorithm follows the chain rule of gradients, it satisfies the requirement of Implementation Invariance in (Sundararajan et al., 2017).

5. Experiments

In our study, we orchestrate an array of experiments encompassing three models, namely Inception-v3 (Szegedy et al., 2016), ResNet-50 (He et al., 2016), and VGG16 (Simonyan & Zisserman, 2014). The focal objective of these designed experiments is to discern the relative efficacy of seven distinct attribution methods, namely IG (Sundararajan et al., 2017), FastIG (FIG) (Hesse et al., 2021), GuidedIG (GIG) (Kapishnikov et al., 2021), BIG (Wang et al., 2022), SaliencyMap (SM) (Simonyan et al., 2013), AGI (Pan et al., 2021), and ISA (our work).

To statistically analyse and evaluate the performance characteristics, we apply the Insertion and Deletion score (Pan et al., 2021). We demonstrate that ISA has better performance compared to other attribution methods.

5.1. Dataset

In the experiment, we employ the widely used ImageNet (Deng et al., 2009) dataset. We randomly select 1000 samples from ImageNet dataset to evaluate the performance of various attribution methods. The sample size is determined based on the guidelines followed by FIA (Wang et al., 2021), NAA (Zhang et al., 2022), and AGI (Pan et al., 2021) experiments.

5.2. Evaluation Metrics

We follow the evaluation metrics used in AGI (Pan et al., 2021), namely the Insertion and Deletion score. The Insertion score measures the extent of output change in the model when pixels are inserted into the input. If we draw a curve that represents the prediction values, the area under the curve (AUC) is then defined as the insertion score. Higher the insertion score, the better the quality of interpretation. Conversely, the Deletion score quantifies the impact on the model's output when pixels are removed from the input. The lower the deletion score, the better the quality of interpretation. However, due to the adversarial nature of neural networks, the Deletion score may not always provide reliable indications (Petsiuk et al., 2018). Thus, the Insertion score offers broader and more informative insights compared to the Deletion score. In addition, we apply the INFD score (Yeh et al., 2019) to analyze the faithfulness of ISA to the underlying model. The lower the INFD score, the more faithful it is to the underlying model.

In-depth analysis Due to the fact that the Insertion score starts from a baseline of inserting crucial features, it measures how much the model output changes when features are inserted into the input. Therefore, the Insertion score is an accumulation of accuracy from the beginning. If these features are indeed significant, the Insertion score will exhibit a rapid increase. On the other hand, the Deletion score starts from the original image and represents the accumulation of accuracy of the remaining parts when features are deleted. Thus, the accuracy calculated by the deletion score is based on the feature deletion process. The model can deduce partial information from the surrounding background, which cannot be controlled for the removed features. At this time, the undeleted features will interfere with the accuracy and affect the effect of the interpretability evaluation. Therefore, the Insertion score serves as a more representative indicator of the performance of attribution algorithms.

5.3. Experiments Setting

We perform the experiments on a platform with a single Nvidia RTX3090 GPU. Meanwhile, we configure the experiment with several critical parameters. Specifically, we set the step size to be 5000, ascent step T_1 and descent step T_2 to be 8 of each, learning rate to 0.002, and S to 1.1.

5.4. Result

Table 1. Insertion and Deletion sore

Model	Method	Deletion score (mean)	Deletion score (AUC)	**Insertion score (mean)**	**Insertion score (AUC)**
Inception-v3	IG	0.0445	0.0426	0.3215	0.3208
	FIG	0.0475	0.0456	0.2029	0.2017
	GIG	0.0363	0.0343	0.3194	0.3187
	BIG	0.0557	0.0538	0.4840	0.4840
	SM	0.0649	0.0631	0.5331	0.5334
	AGI	0.0676	0.0658	0.6249	0.6256
	ISA	0.0542	0.0523	**0.7335**	**0.7346**
ResNet-50	IG	0.0302	0.0283	0.1467	0.1454
	FIG	0.0342	0.0324	0.1078	0.1063
	GIG	0.0210	0.0191	0.1463	0.1450
	BIG	0.0485	0.0467	0.2911	0.2905
	SM	0.0585	0.0567	0.3160	0.3154
	AGI	0.0532	0.0515	0.5133	0.5136
	ISA	0.0529	0.0512	**0.6073**	**0.6065**
VGG16	IG	0.0249	0.0232	0.0973	0.0959
	FIG	0.0288	0.0270	0.0809	0.0793
	GIG	0.0191	0.0173	0.1040	0.1025
	BIG	0.0390	0.0372	0.2274	0.2266
	SM	0.0434	0.0417	0.2710	0.2703
	AGI	0.0459	0.0442	0.4303	0.4304
	ISA	0.0440	0.0423	**0.5085**	**0.5082**

As shown in Table. 1 and Table. 2, the ISA method achieves the best attribution results with the highest Insertion score and the lowest INFD score, which indicates that the ISA method outperforms other attribution methods for the attribution task. Specifically, the increase in Insertion score of the ISA method compared to other attribution methods is relatively large, with average increases of 0.3206 and 0.3538, and 0.3074 on Inception-v3, ResNet-50, and VGG16, respectively, indicating that the method has significantly improved the attribution performance.

Table 2. INFD score

Method	INFD Score		
	Inception-v3	ResNet-50	VGG16
IG	88.39	87.73	139.72
FIG	173.09	161.32	301.12
GIG	89.70	61.43	94.65
BIG	13.29	1.77	17.34
SM	26.79	5.13	19.09
AGI	4.98	0.84	3.36
ISA	**4.82**	**0.83**	**0.56**

The comparison between ISA and GradCAM and the non-CNN comparative experiments (ViT-B/16) (Dosovitskiy et al., 2020) can be found in the **Appendix. D** and **Appendix. E**.

5.5. Attribution complexity analysis

Regarding the attribution efficiency (time cost), it is typical to evaluate via the number of forward and back propagation times, such as in AGI(Pan et al., 2021). Following this way, we firstly consider the computational complexity of ISA is:

$$\left\lceil \frac{n}{k} \right\rceil \cdot (T_1 + T_2) \tag{12}$$

where $\lceil \cdot \rceil$ denotes the rounding up function. In our experiments, $T_1 = 8$ and $T_2 = 8$ represent the steps for gradient ascent and gradient descent. For the total feature number n $(224 \times 224 \times 3)$, we will attribute as large as possible k features at a time (in our submission, $k = 5000$). Eventually, $\left\lceil \frac{n}{k} \right\rceil$ is about 30. For the AGI method, the computational complexity becomes $k \cdot m$, where k is the number of false classes sampled, and m is the maximum number of iterations. In (Pan et al., 2021), AGI takes $k = 20, m = 20$ on Inception-v3 respectively.

Thus, our method demonstrates a huge performance improvement whilst attributing $k = 5000$ per attribution, although the time takes slightly longer than AGI (AGI is propagated 400 times and ISA is propagated 480 times). Overall, we consider the computational complexity of our algorithm to be reasonably comparable despite the iterative attribution nature, which could be very close to the AGI method. We posit that, despite the efficiency gap arising from a slightly higher number of gradient propagations, the performance breakthrough achieved by our algorithm is a noteworthy outcome. This is also similar to why we prefer

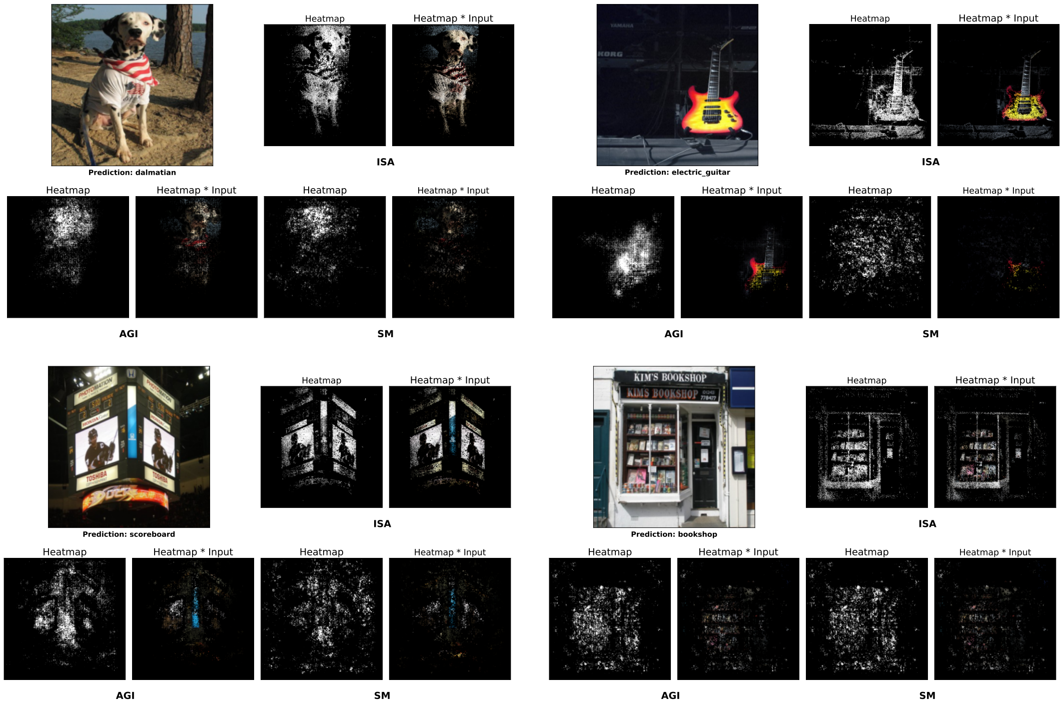

Figure 2. Attribution visualization for *Scoreboard* Image using ISA, AGI, and SM (**Appendix. G** for additional results)

to use diffusion models to generate high quality samples. Although we know that diffusion models usually take longer than non-diffusion models, we are actually willing to bear these efficiency reductions to obtain higher performance.

5.6. Ablation Study

In order to validate the efficacy of the ISA method, a series of ablation study is conducted using the Inception-v3 model. These experiments aim to investigate the impact of various parameters on the model's performance. Specifically, we explore the effects of combining parameters of ascent step T_1 and descent step T_2, the effects of parameter step size, the effects of the parameter learning rate, and the effects of the parameter S. In the **Appendix. F**, we additionally provide the ablation performance of ISA on VGG_16 and ResNet-50 models.

5.6.1. The effects of ascent step T_1 and descent step T_2

In this section, we compare the effects of two different approaches, gradient ascent and gradient descent, on the method's attribution performance. To acieve this, we set the following parameters: step size at 5000, ascent and descent steps at 8, learning rate at 0.004, and S at 1.3. Three sets of experiments are conducted: gradient descent only, gradient ascent only, and simultaneous gradient descent and ascent. The combinations of these parameters are summarized in Table 3. In the gradient descent-only and gradient ascent-only experiments, we observe similar values for the Insertion score and Deletion score. This similarity indicates that the attribution effects of these two methods are comparable. However, when gradient descent and ascent are performed simultaneously, the experimental results exhibit a higher Insertion score and a lower Deletion score. This suggests that the parameter combination used in the simultaneous approach outperforms the comparison experiments in terms of attribution effect. The higher the T_1 and T_2 values represent the deeper the exploration of the input space. This effectively demonstrates how altering these two parameters can influence the exploration of the input space through gradient ascent and descent, thereby impacting the performance of attribution.

Table 3. Insertion sore and deletion score with different gradient parameters

	T_1	T_2	Insertion score	Deletion score
Gradient descent only	0	8	0.7042	0.0715
Gradient ascent only	8	0	0.7055	0.0739
Gradient descent and ascent	8	8	**0.7346**	**0.0523**

5.6.2. The effects of step size k

In this experiment, we conduct a comparison of the effect of different step sizes on the performance of ISA. Initially, we set ascent step T_1 and descent step T_2 to 8, learning

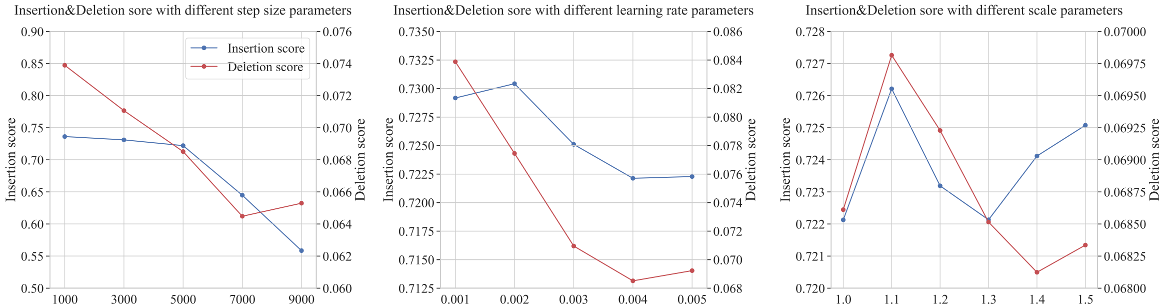

Figure 3. Insertion and Deletion score comparison of ISA

rate to 0.004, and S to 1.3. We tested step sizes of 1000, 3000, 5000, 7000, and 9000, respectively. The results are presented in Figure 3.

From the figure, we can observe a decreasing trend in both the Insertion score and Deletion score as the step size increases. Specifically, when the step size is set to 1000, the Insertion score reaches its maximum value across all experiment results. Conversely, as the step size is increased to 9000, the Insertion score will be the lowest, accompanied by an increase in the Deletion score. A profound analysis of this phenomenon reveals that the parameter step size means the number of unimportant attribution values to be removed in each iteration. A larger step size means more attribution values are removed in each iteration. We found that when step size is 5000, the algorithm achieves the best results. We believe that when step size is too low, the model may not be able to fully capture the contribution of different attribution values to the model's decision-making behavior. When step size is too high, noise may be introduced, some of which may be information irrelevant to the model, leading to inaccuracy in model interpretation.

5.6.3. The effects of learning rate η

In this experiment, we conduct a comparison of the attribution effects of ISA using different learning rates. The parameters are set as follows: ascent step T_1 and descent step T_2 at 8, step size at 5000, and S at 1.3. Subsequently, we evaluate the performance of ISA at learning rates of 0.001, 0.002, 0.003, 0.004, and 0.005, respectively.

The results are presented in Figure 3. As depicted in the figure, we can see that ISA achieves the highest Insertion score and competitively low Deletion score when the learning rate is 0.002. Both the Insertion score and Deletion score decline dramatically while the learning rates is increased. This is because the learning rate affects the exploration process of the input space by gradient ascent and gradient descent. For gradient ascent, a too high learning rate may lead to over-exploration of the input space, making the interpretation too unstable. For gradient descent, a too high learning rate may cause the interpretation results to be too sensitive.

5.6.4. The effects of scaling factor S

In this section, we conduct a performance comparison of ISA using different scales. The experimental setup involves setting ascent step T_1 and descent step T_2 to 8, step size to 5000, and learning rate to 0.04. We then test six different S: 1.0, 1.1, 1.2, 1.3, 1.4, and 1.5.

The results are depicted in Figure 3. From the figure, we can see that both the Insertion score and Deletion score exhibit similar trends as S increases. Therefore, we select the parameter with the highest Insertion score as the optimal choice. In this case, ISA achieve the best performance with a scale of 1.3. We posit that if the value of the parameter is excessively large (approaching 2), the relative importance of the attribution values removed in successive iterations becomes more proximate, thereby exhibiting an over-intermingling effect. Conversely, if the value is too diminutive, signifying under-intermingling, a higher degree of precision in estimation is necessitated. In such instances, the attribution value removed in each iteration may not be as optimal as the attribution value removed in the subsequent iteration.

6. Conclusion

In this paper, we propose a novel attribution method, termed Iterative Search Attribution (ISA), to better interpret deep neural networks. Specifically, we consider that both gradient ascent and gradient descent are important for the exploration of feature importance. The relatively unimportant features for the model are clipped to achieve more accurate attribution results. Comprehensive experimental results show that our method has superior performance for image recognition interpretability tasks compared to other state-of-the-art baselines. Given the limitation that we only explore the iterative attribution value by removing features in equal amounts, we will investigate the performance of our algorithm by varying the removal of features in unequal amounts in future work.

Impact Statement

This paper presents work whose goal is to advance the field of explainable artificial intelligence, in particularly the proposed method ISA significantly enhances the interpretability of deep neural networks, promoting trust and transparency in AI applications across critical fields such as healthcare and finance. By providing more accurate attributions, ISA aids in informative decision-making and advances research in explainable AI. While there are many potential societal consequences of our work, we understand it may be misused to exploit AI system vulnerabilities, pose privacy concerns by revealing sensitive data, and lead to an over-reliance on interpretability. To mitigate these risks, controlled access to models, robust data protection measures, and comprehensive user training are essential. These strategies ensure that the benefits of ISA are maximized while minimizing potential negative impacts, contributing to the responsible advancement of AI technologies.

References

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

Collobert, R. and Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pp. 160–167, 2008.

Datta, A., Sen, S., and Zick, Y. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *2016 IEEE symposium on security and privacy (SP)*, pp. 598–617. IEEE, 2016.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Erion, G., Janizek, J. D., Sturmfels, P., Lundberg, S. M., and Lee, S.-I. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature machine intelligence*, 3(7):620–631, 2021.

Esteva, A., Chou, K., Yeung, S., Naik, N., Madani, A., Mottaghi, A., Liu, Y., Topol, E., Dean, J., and Socher, R. Deep learning-enabled medical computer vision. *NPJ digital medicine*, 4(1):5, 2021.

Fong, R. C. and Vedaldi, A. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*, pp. 3429–3437, 2017.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hesse, R., Schaub-Meyer, S., and Roth, S. Fast axiomatic attribution for neural networks. *Advances in Neural Information Processing Systems*, 34:19513–19524, 2021.

Jabbar, R., Al-Khalifa, K., Kharbeche, M., Alhajyaseen, W., Jafari, M., and Jiang, S. Real-time driver drowsiness detection for android application using deep neural networks techniques. *Procedia computer science*, 130: 400–407, 2018.

Kapishnikov, A., Venugopalan, S., Avci, B., Wedin, B., Terry, M., and Bolukbasi, T. Guided integrated gradients: An adaptive path method for removing noise. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5050–5058, 2021.

Li, X., Pan, D., Li, C., Qiang, Y., and Zhu, D. Negative flux aggregation to estimate feature attributions. *arXiv preprint arXiv:2301.06989*, 2023.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Maas, A. L., Qi, P., Xie, Z., Hannun, A. Y., Lengerich, C. T., Jurafsky, D., and Ng, A. Y. Building dnn acoustic models for large vocabulary speech recognition. *Computer Speech & Language*, 41:195–213, 2017.

Ozcan, A., Catal, C., Donmez, E., and Senturk, B. A hybrid dnn–lstm model for detecting phishing urls. *Neural Computing and Applications*, pp. 1–17, 2021.

Pan, D., Li, X., and Zhu, D. Explaining deep neural network models with adversarial gradient integration. In *Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

Pan, J., Liu, C., Wang, Z., Hu, Y., and Jiang, H. Investigation of deep neural networks (dnn) for large vocabulary continuous speech recognition: Why dnn surpasses gmms in acoustic modeling. In *2012 8th International Symposium on Chinese Spoken Language Processing*, pp. 301–305. IEEE, 2012.

Pathak, A. R., Pandey, M., and Rautaray, S. Application of deep learning for object detection. *Procedia computer science*, 132:1706–1717, 2018.

Petsiuk, V., Das, A., and Saenko, K. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018.

Rajendran, S. and Topaloglu, U. Extracting smoking status from electronic health records using nlp and deep learning. *AMIA Summits on Translational Science Proceedings*, 2020:507, 2020.

Ribeiro, M. T., Singh, S., and Guestrin, C. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

Rogers, E. M. A prospective and retrospective look at the diffusion model. *Journal of health communication*, 9(S1): 13–19, 2004.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Shrikumar, A., Greenside, P., and Kundaje, A. Learning important features through propagating activation differences. In *International conference on machine learning*, pp. 3145–3153. PMLR, 2017.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Wang, H., Wang, Z., Du, M., Yang, F., Zhang, Z., Ding, S., Mardziel, P., and Hu, X. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 24–25, 2020.

Wang, Z., Guo, H., Zhang, Z., Liu, W., Qin, Z., and Ren, K. Feature importance-aware transferable adversarial attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7639–7648, 2021.

Wang, Z., Fredrikson, M., and Datta, A. Robust models are more interpretable because attributions look normal. In *International Conference on Machine Learning*, pp. 22625–22651. PMLR, 2022.

Yeh, C.-K., Hsieh, C.-Y., Suggala, A., Inouye, D. I., and Ravikumar, P. K. On the (in) fidelity and sensitivity of explanations. *Advances in Neural Information Processing Systems*, 32, 2019.

Zhang, J., Wu, W., Huang, J.-t., Huang, Y., Wang, W., Su, Y., and Lyu, M. R. Improving adversarial transferability via neuron attribution-based attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14993–15002, 2022.

Zhu, Z., Chen, H., Zhang, J., Wang, X., Jin, Z., Xue, M., Zhu, D., and Choo, K.-K. R. Mfaba: A more faithful and accelerated boundary-based attribution method for deep neural networks. *arXiv preprint arXiv:2312.13630*, 2023.

A. Detailed proofs of the axiom of Sensitivity

Firstly, during the iterative process, the changes in the gradient along the integration path are captured by the original input information. Furthermore, it is not retroactive since feature values in previous iterations are unchanged in subsequent iterations. Therefore, the attribution result must be non-zero, which meets the definition of sensitivity. Here is the mathematical proof.

We first use the first-order Taylor approximation to expand the loss function and combine the information for the path from x_0 to x_T.

$$\begin{aligned} L\left(x_t\right) &= L\left(x_{t-1}\right) \pm \frac{\partial L\left(x_{t-1}\right)}{\partial x_{t-1}}\left(x_t - x_{t-1}\right) + \varepsilon \\ \sum_{t=1}^{T} L\left(x_t\right) &= \sum_{t=0}^{T-1} L\left(x_t\right) \pm \sum_{t=0}^{T-1} \frac{\partial L\left(x_t\right)}{\partial x_t}\left(x_{t+1} - x_t\right) \\ A = L\left(x_T\right) - L\left(x_0\right) &= \pm \sum_{t=0}^{T-1} \frac{\partial L\left(x_t\right)}{\partial x_t}\left(x_{t+1} - x_t\right) \\ &= \pm \sum_{t=0}^{T-1} \frac{\partial L\left(x_t\right)}{\partial x_t} \cdot \triangle x_t = \pm \int_T \triangle x_t \cdot \frac{\partial L\left(x_t\right)}{\partial x_t} \mathrm{d}t \end{aligned} \tag{13}$$

Here ϵ is omitted due to the principle of higher-order Taylor expansions. L represents the loss function. x_t represents the input of the t-th iteration. We can know that as long as the loss function of the model changes, the attribution result will definitely be non-zero.

Since our ISA algorithm combines gradient ascent and gradient descent to explore the input space, the ISA attribution path can be expanded as follows:

- For the input space exploration of gradient ascent, the attribution path of ISA is x_0, x_1,..., x_{T_1}.
- For the input space exploration of gradient descent, the attribution path of ISA is x_0, x_1,..., x_{T_2}.

For the gradient ascent process, we can get the following formula:

$$\begin{aligned} A_a = L\left(x_{T_1}\right) - L\left(x_0\right) &= \sum_{t=0}^{T_1-1} \frac{\partial L\left(x_t\right)}{\partial x_t}\left(x_{t+1} - x_t\right) \\ &= \sum_{t=0}^{T_1-1} \frac{\partial L\left(x_t\right)}{\partial x_t} \cdot \triangle x_t = \int_{T_1} \triangle x_t \cdot \frac{\partial L\left(x_t\right)}{\partial x_t} \mathrm{d}t \end{aligned} \tag{14}$$

We define $\triangle L_a = L(x_{T_1}) - L(x_0) = c_1$. Here c_1 is a constant with a positive sign. Thus, we get $\bar{A}_a = \frac{A_a}{\triangle L_a}$. According to Eq. 2, $\bar{A}_a$ can be expressed as:

$$\begin{aligned} \bar{A}_a = \frac{A_a}{\triangle L_a} &= \frac{\sum_{t=0}^{T_1-1} \frac{\partial L(x_t)}{\partial x_t}\left(x_{t+1} - x_t\right)}{L(x_{T_1}) - L(x_0)} \\ &= \frac{\sum_{t=0}^{T_1-1} \frac{\partial L(x_t)}{\partial x_t} \cdot \triangle x_t}{L(x_{T_1}) - L(x_0)} = \frac{1}{c_1} \int_{T_1} \triangle x_t \cdot \frac{\partial L\left(x_t\right)}{\partial x_t} \mathrm{d}t = 1 \end{aligned} \tag{15}$$

Obviously the attribution result is normalized to 1, which satisfies sensitivity. It is worth noting that since c_1 is a constant with a positive sign, during gradient ascent, we can use $L' = \frac{L}{c_1}$ to replace the loss function L in Eq. 3, so sensitivity is also satisfied at this time.

For the gradient descent process, similarly, we can get the following formula:

$$\begin{aligned} A_d = L\left(x_{T_2}\right) - L\left(x_0\right) &= -\sum_{t=0}^{T_2-1} \frac{\partial L\left(x_t\right)}{\partial x_t}\left(x_{t+1} - x_t\right) \\ &= -\sum_{t=0}^{T_2-1} \frac{\partial L\left(x_t\right)}{\partial x_t} \cdot \triangle x_t = -\int_{T_2} \triangle x_t \cdot \frac{\partial L\left(x_t\right)}{\partial x_t} \mathrm{d}t \end{aligned} \tag{16}$$

We define $\triangle L_d = L(x_{T_2}) - L(x_0) = c_2$. Here c_2 is a constant with a negative sign. Thus, we get $\bar{A}_d = \frac{A_d}{\triangle L_d}$. According to Eq. 4, $\bar{A}_d$ can be expressed as:

$$\begin{aligned} \bar{A}_d = \frac{A_d}{\triangle L_d} &= \frac{-\sum_{t=0}^{T_2-1} \frac{\partial L(x_t)}{\partial x_t}\left(x_{t+1} - x_t\right)}{L(x_{T_2}) - L(x_0)} \\ &= \frac{\sum_{t=0}^{T_2-1} \frac{\partial L(x_t)}{\partial x_t} \cdot \triangle x_t}{L(x_0) - L(x_{T_2})} = \frac{1}{-c_2} \int_{T_2} \triangle x_t \cdot \frac{\partial L\left(x_t\right)}{\partial x_t} \mathrm{d}t = 1 \end{aligned} \tag{17}$$

Obviously the attribution result is normalized to 1, which satisfies sensitivity. It is worth noting that since c_2 is a constant with a negatove sign, during gradient descent, we can use $L'' = \frac{L}{-c_2}$ to replace the loss function L in Eq. 5, so sensitivity is also satisfied at this time.

Finally, we make a balance between gradient ascent and descent by combining them in the following formula:

$$\bar{A} = \bar{A}_a + \bar{A}_d = \int_{T_1} \triangle x_t \cdot \frac{\partial L'\left(x_t\right)}{\partial x_t} \mathrm{d}t + \int_{T_2} \triangle x_t \cdot \frac{\partial L''\left(x_t\right)}{\partial x_t} \mathrm{d}t = 2 \tag{18}$$

We get $\frac{\bar{A}}{2} = \frac{\int_{T_1} \triangle x_t \cdot \frac{\partial L'(x_t)}{\partial x_t} \mathrm{d}t + \int_{T_2} \triangle x_t \cdot \frac{\partial L''(x_t)}{\partial x_t} \mathrm{d}t}{2} = 1$, which also satisfies sensitivity.

B. Proof of Theorem 4.2

Assume x_i and x_j are the output features of x_0 after one-step gradient ascent, where$\{\triangle x_i, \triangle x_j\} \in R^n$. Our ***Theorem 4.2*** can be transformed into proving $L(x_0 + \triangle \bar{x_i}) \geq L(x_0 + \triangle \bar{x_j})$ if $A_a^i \geq A_a^j$. Due to the first-order Taylor expansion, we can get:

$$L(x_0 + \triangle \bar{x_i}) = L(x_0) + \underbrace{\triangle x_i \cdot \frac{\partial L(x_i)}{\partial x_i}}_{A_a^i} + o \tag{19}$$

$$L(x_0 + \triangle \bar{x_j}) = L(x_0) + \underbrace{\triangle x_j \cdot \frac{\partial L(x_j)}{\partial x_j}}_{A_a^j} + o \tag{20}$$

If $A_a^i \geq A_a^j$, obviously $L(x_0 + \triangle \bar{x_i}) \geq L(x_0 + \triangle \bar{x_j})$. The case of multi-step gradient ascent here also satisfies ***Theorem 4.2***.

Similarly, if $A_d^i \leq A_d^j$, then $L(x_0 + \triangle \bar{x_i}) \geq L(x_0 + \triangle \bar{x_j})$. Since this is the case of gradient descent, the larger loss function $L(x_0 + \triangle \bar{x_i})$ means the lower the contribution to enhanced model decision-making.

C. Iterative error analysis

Assume that the feature matrix $M = [x] = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ is composed of block matrices M_1 and M_2. Here $M_1 = [x_1]$,$M_2 = [x_2]$. At this time $x_1 \in R^{n_1}$, $x_2 \in R^{n_2}$, $n_1 + n_2 = n$. Then for feature variation $\triangle x = \begin{bmatrix} \triangle x_1 \\ \triangle x_2 \end{bmatrix}$, we have the feature matrix

$x' = x + \triangle x = \begin{bmatrix} x_1 + \triangle x_1 \\ x_2 + \triangle x_2 \end{bmatrix}$ after gradient ascent. In order to obtain the attribution to x_2, we can perform the following calculation:

$$A_{x_2} = \triangle x_2 \cdot \frac{\partial L(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix})}{\partial x_2} \tag{21}$$

We then can perform Taylor expansion to $\frac{\partial L(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix})}{\partial x_2}$:

$$A_{x_2} = \triangle x_2 \cdot \left[\frac{\partial L(\begin{bmatrix} 0 \\ x_2 \end{bmatrix})}{\partial x_2} + \triangle x_1 \frac{\partial^2 L(\begin{bmatrix} 0 \\ x_2 \end{bmatrix})}{\partial x_2 \partial x_1} + o \right] = \triangle x_2 \cdot \frac{\partial L(\begin{bmatrix} 0 \\ x_2 \end{bmatrix})}{\partial x_2} + \triangle x_2 \triangle x_1 \cdot \frac{\partial^2 L(\begin{bmatrix} 0 \\ x_2 \end{bmatrix})}{\partial x_2 \partial x_1} + o \cdot \triangle x_2 \tag{22}$$

If we think that the features in the block matrix M_1 are unimportant and change them to 0, then the attribution to x_2 is:

$$A'_{x_2} = \triangle x_2 \cdot \frac{\partial L(\begin{bmatrix} 0 \\ x_2 \end{bmatrix})}{\partial x_2} \tag{23}$$

Since o represents higher-order infinitesimal, when feature x_1 is unseen to the model, the attribution error for x_2 is approximately:

$$Error = \triangle x_2 \triangle x_1 \cdot \frac{\partial^2 L(\begin{bmatrix} 0 \\ x_2 \end{bmatrix})}{\partial x_2 \partial x_1} \tag{24}$$

In the process of continuous iteration, the block matrix M_2 can be continued to be divided into blocks, and the error calculation during gradient descent is the same.

D. Comparative experiment on NLP

Table 4. Interpretable performance of different methods on LSTM model

Method	IG	FIG	SG	DeepLIFT	SM	BIG	AGI	ISA
INS	0.6825	0.4934	0.6541	0.6658	0.7646	0.747	0.6691	0.8316
DEL	0.5016	0.6657	0.5746	0.4932	0.5666	0.687	0.5923	0.5477

E. The comparison between ISA and GradCAM

Table 5. Comparison between ISA and GradCAM

	Inception-v3		ResNet-50		VGG16	
Method	Insertion	Deletion	Insertion	Deletion	Insertion	Deletion
GradCAM	0.5798	0.1594	0.3417	0.1231	0.4545	0.1092
ISA	**0.7293**	**0.0745**	**0.6043**	**0.0619**	**0.5111**	**0.0504**

F. Comparative experiment on ViT

G. Guidelines of hyperparameter selection

We further elaborate the detailed ablation experiments on the scale parameter, learning rate, feature removal step size, and gradient ascent & descent steps as discussed in Section 5.6.

Table 6. Comparative experiment on ViT

	vit_b_16	
Method	Insertion	Deletion
Fast-IG	0.2682	0.0883
Saliency Map	0.3374	0.098
IG	0.3782	0.068
GIG	0.366	0.06306
AGI	0.5014	0.0849
BIG	0.4704	0.1132
ISA	**0.6715**	0.1153

In Section 5.6.4, we posit that the scale parameter adjusts the importance between the attribution values removed in adjacent iterations. We found that, if the scale parameter is too large, it may result in the importance of the removed attribution values too close, making it too difficult to distinguish the effect of each iteration. In the other way, if it is too low, it may affect the removal result of attribution values being not precise, requiring more iterations to find the optimal solution. Thus, we consider a moderate level of 0.3 is a value for the scale parameter, which can largely balance the performance of the algorithm.

We want to emphasize that, even without the scaling parameter, our algorithm still achieves the best performance. This can be seen in the third subgraph of Figure 3, where a Scale of 1 is equivalent to not applying any Scale operation. If we initially work on new tasks, a Scale of 1 can be selected to obtain the attribution result, followed by the trials of other Scales subsequently. Since calculating insertion and deletion scores for a small number of samples is efficient, we recommend such an evaluation method on new tasks to validate the effectiveness of Scale.

In Section 5.6.3, we observed that performance drops sharply with an increase in the learning rate. Our reason is that a high learning rate leads to an unstable exploration of the input space during gradient ascent and overly sensitive exploration during gradient descent. In this way, we suggest a relatively low value of learning rate at 0.002 will achieve promising performance.

In Section 5.6.2, we found that while the highest insertion score is achieved when the feature removal step size is 1000, it also incurs the largest computational cost and the highest deletion score. When the step size is 9000, the insertion score is the lowest, and the deletion score starts to rise. Therefore, we believe that both too large and too small step sizes fail to achieve promising effects. An in-depth analysis reveals that if the step size is too low, the model may not fully capture the contribution of different attribution values to the model's decision-making behavior, requiring additional iterations. If the step is too high, it may introduce noise unrelated to the model, leading to inaccurate model explanations. In practice, we set the step size to a moderate level of 5000 to balance the trade-off between model efficiency and performance.

In Section 5.6.1, we argue that gradient ascent & descent steps are responsible for the depth of exploration in the input space. Clearly, a higher number of exploration steps corresponds to higher performance. Additionally, we conducted ablation studies on either performing gradient ascent or descent separately, or performing both in combination. The results showed that combining both achieves best performance. Considering the efficiency of the algorithm, we set both gradient ascent & descent steps to 8 in practice.

H. The ablation study of ISA on ResNet_50 and VGG_16 models

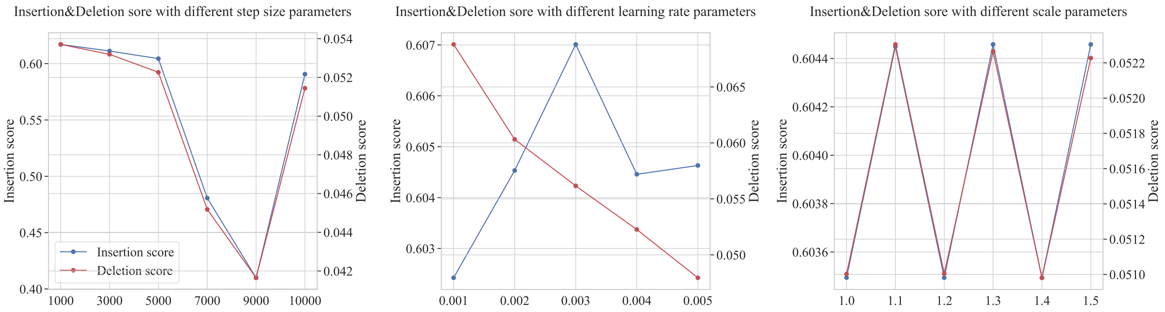

Figure 4. The ablation study of ISA on ResNet_50

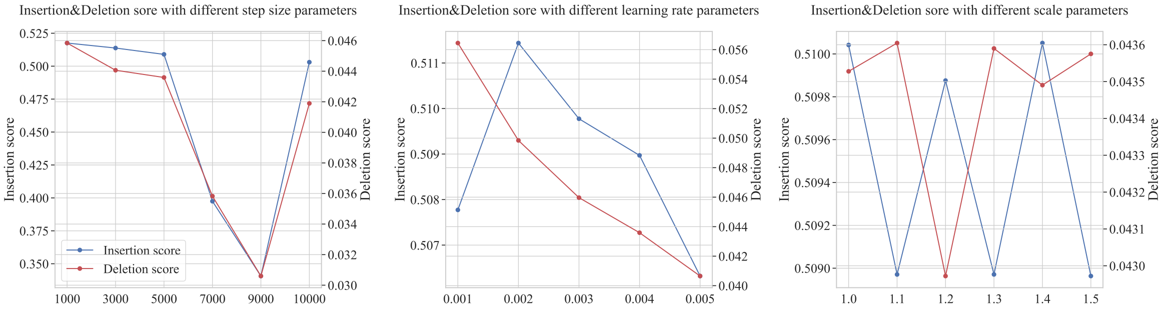

Figure 5. The ablation study of ISA on VGG_16

I. Additional visualization

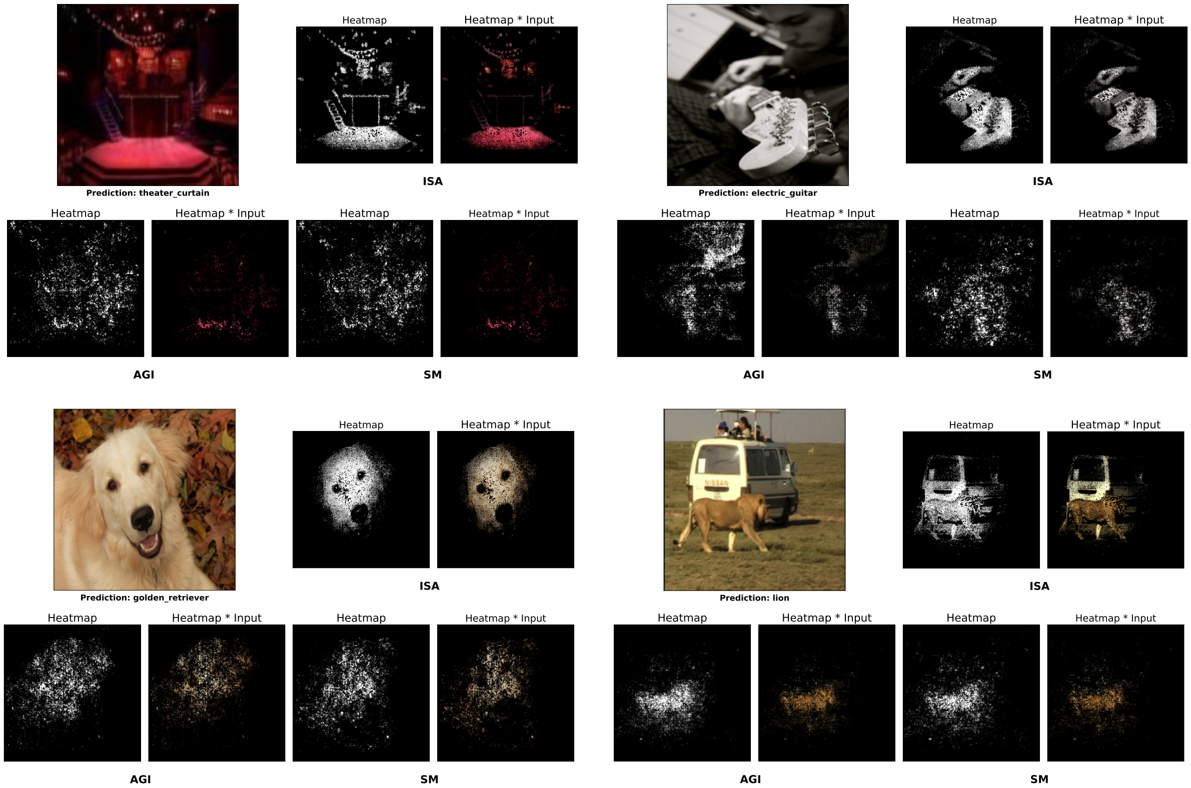

Figure 6. Additional visualization for *Scoreboard* Image using ISA, AGI, and SM

J. Pseudocode

Algorithm 1 Iterative Search Attribution

```
Input: Original input feature $x_0$, parameter matrix $W$, step size $\eta_1$, step size $\eta_2$, ascent step $T_1$, descent step $T_2$, feature
    removal number $k$, integration step $\Gamma$, loss funtion $L$, Scaling factor $S$, mask $m$
Output: $A^*$
Initial: $A^* = 0$, $A_a = 0$, $A_d = 0$
for $\gamma$ in range $\Gamma$ do
    for $t = 0, 1, ..., T_1$ do
        $x_{t+1} = x_t + \eta_1 \cdot sign(\frac{\partial L(x_t)}{\partial x_t})$
        $A_a = A_a + \eta_1 \cdot sign(\frac{\partial L(x_t)}{\partial x_t}) \cdot \frac{\partial L(x_t)}{\partial x_t}$
    end for
    for $t = 0, 1..., T_2$ do
        $x_{t+1} = x_t - \eta_2 \cdot sign(\frac{\partial L(x_t)}{\partial x_t})$
        $A_d = A_d - \eta_2 \cdot sign(\frac{\partial L(x_t)}{\partial x_t}) \cdot \frac{\partial L(x_t)}{\partial x_t}$
    end for
    $\Delta L_a = L\left(x_{T_1}\right) - L\left(x_0\right)$, $\Delta L_d = L\left(x_{T_2}\right) - L\left(x_0\right)$
    $\bar{A}_a = \frac{A_a}{\Delta L_a}$, $\bar{A}_d = \frac{A_d}{\Delta L_d}$
    $\bar{A} = \bar{A}_a + \bar{A}_d$
    $\bar{a_\gamma} = min_k(\bar{A_\gamma})$, remove the features corresponding to the $k$ minimum attribution values
    $\bar{a_\gamma} = \frac{\bar{a_\gamma} - min(\bar{a_\gamma})}{max(\bar{a_\gamma}) - min(\bar{a_\gamma})}$
    $\bar{a_\gamma} = \bar{a_\gamma} \cdot S + \gamma$
    $A^* = A^* + \bar{a_\gamma}$
end for
```